

WHAT IS CLAIMED IS:

1. A method for detecting memory corruption comprising:
performing a data write to a predefined memory segment;
after said write, comparing a redundant portion at the end of said segment to an initial header portion of said segment;
if said comparing indicates that data has been written into said redundant portion,
signaling an overwrite error.
2. A method according to claim 1 wherein said signaling occurs in real-time and can detect and isolate a process causing said erroneous overwrite.
3. A method according to claim 1 wherein said initial portion includes a memory segment header and includes a pointer to a next segment in a linked list.
4. A method according to claim 1 wherein said memory segment is a buffer in a buffer system comprised of a plurality of buffers.
5. A method according to claim 1 further comprising:
if said comparing indicates a difference between said initial portion and said redundant portion, validating one of said portions and copying a validated portion to the location of a non-validated portion.
6. A method according to claim 1 further comprising:
dumping the call stack to isolate the error to the routine that caused the rewrite.
7. A method for allocating memory segments so as to provide for detecting memory corruption comprising:
at initial memory segment allocation, providing a redundant portion at the logical end of a memory segment, said redundant portion containing data derived from an initial header portion of said segment; and
providing a verify memory routine for comparing said redundant portion to said initial portion to determine if an erroneous overwrite has occurred.

8. A method for detecting and correcting memory corruption comprising:
upon accessing a memory segment, comparing a redundant portion at the end of said segment to an initial header portion or said segment;
if said comparing indicates a difference between said segments, determining which portion is valid; and
from said valid portion, reconstructing said invalid portion.
9. A method according to claim 8 wherein said comparing determines if said initial portion and said redundant portion are identical.
10. A method according to claim 8 wherein said determining which portion is valid comprises for each portion comparing stored data in that portion against a check sum value.
11. A method according to claim 8 wherein said determining which portion is valid comprises for each portion determining if a pointer in that portion is within a memory pool range.
12. A buffer system able to detect buffer corruption comprising:
a plurality of buffers (110, 120, 130) with logically contiguous addressing;
a least one buffer comprising an initial portion (110a) and a final portion (110b);
a detector for determining that a write has been performed on said at least one buffer;
a comparater for comparing said initial portion and said final portion after a write; and
a signaler for sending an alert when said comparater indicates an erroneous overwrite has occurred wherein said signaler can cause stack to be dumped and other debug information to be collected.
13. An information processing system comprising:
a process space (500) containing a plurality of executable logic components;
buffer allocation logic (510) able to establish a buffer structure in a memory space (550);
buffer write logic (520) able to write data to buffers; and

buffer verify logic (530) able to determine if an overwrite has occurred by comparing a header portion and an initial portion of a written to buffer.

14. The device according to claim 13 further comprising:
repair logic (540) able to repair an overwritten portion.
15. The device according to claim 13 further comprising:
a buffer structure (560) comprising a series of linked memory segments arranged in a memory space.
16. An apparatus according to claim 12 wherein said apparatus is embodied into a fixed media containing logic instructions that when loaded into appropriately configured computer systems will cause the system to embody said buffer system.